

IN THE CLAIMS

What is claimed is:

1. (Currently Amended) A method for remote invocation of an object in a base object library via a remote access object library comprising:
 - invoking, via a client application interface, an API object reference in the remote access object library;
 - identifying a corresponding native object to the invoked API object reference in the base object library;
 - instantiating ~~an API object reflective of the identified native object as an API object~~ in a client object space, ~~the API object selectively exposing the corresponding native object~~; and
 - maintaining a link between the instantiated API object and the corresponding native object, the link providing a dynamic reflection of the native object in the API object, maintaining the link further comprising referencing, in a realtime manner, the native object in response to operations to the instantiated API object, such that the operations produce a nonduplicative, atomic result in the native object via the instantiated API object.
2. (Original) The method of claim 1 wherein instantiating further comprises:
 - copying, in a base application object space, the identified native object, the native object having attributes;
 - identifying, for the attributes of the native object, corresponding attributes in the copied object; and
 - populating the corresponding attributes in the instantiated API object.
3. (Original) The method of claim 2 wherein populating further comprises identifying attributes of the native object and determining, from metadata defining the remote access object library, attribute values of the corresponding instantiated API object.

4. (Canceled)

5. (Original) The method of claim 1 further comprising traversing objects in the remote access object library, the traversing comprising:

invoking a traversal method in the API object in the remote access object space;
identifying, via an object identifier, the corresponding native object in the base application object space;

determining a related object associated with the native object in the base object library;

receiving, via a traversal service and the object identifier, an instantiation of the related object, the related object corresponding to the association to the native object in the base object library and the instantiation received from a copied object of the related object, the instantiation maintaining a dynamic link to the related object.

6. (Original) The method of claim 5 wherein identifying via an object identifier is operable to uniquely identify an instantiation of an object in an object space.

7. (Original) The method of claim 1 wherein the base object library further comprises exposed objects and local objects, each of the exposed objects having a corresponding native object in the object model, the exposed object operable to provide similar operations via the remote application interface as the corresponding native object, and further wherein invoking includes determining an exposure attribute indicative of whether the invoked object is an exposed object.

8. (Original) The method of claim 1 further comprising providing the remote access object library and the client application interface comprising:

identifying templates corresponding to object types, the object types corresponding to operations for providing the remote API;

defining metadata for each of the objects for exposure in the remote API, the metadata identifying, for each of the exposed objects, runtime behavior of the object; and

building, via an API object generator, the exposed objects for invocation by the client application.

9. (Original) The method of claim 8 further comprising identifying, in the base object model, native objects for exposure as exposed objects in the remote access object library.

10. (Original) The method of claim 8 wherein defining metadata further comprises defining attributes for the object in the remote access object library.

11. (Original) The method of claim 8 wherein the base object library defines a native application for providing managed information network services to users.

12. (Original) The method of claim 11 wherein the native application is a storage area network management application having a database of storage area network management information, the storage area network management application operable to manipulate agents corresponding to manageable entities, and wherein the remote access object library is a toolkit operable to provide API entry points into the storage area network management application in a nonintrusive manner for interrogating the agents and corresponding database.

13. (Original) The method of claim 1 further comprising:
modifying the base object library; and
remapping the links between the API objects and a new corresponding native object resulting from the modifying, wherein the client application interface remains unmodified such that the API object remain executable without rebuilding.

-5-

14. (Currently Amended) A method of defining and deploying a remote access object library corresponding to a base application object model comprising:

identifying a subset of objects in the base application object model as exposure objects for inclusion the remote access object library;

defining a client application interface accessible to a client application operable to provide accessibility references for the included objects;

defining, via a requirements instance manager, metadata indicative of attributes for objects in the remote access object library;

defining, via an architecture mapper, object templates corresponding to object types in the base application object model, defining the object templates further comprising:

identifying object classes for the object types in the base application object model; ~~and~~

identifying, for each of the object types, attributes operable for definition in the corresponding object in the remote access object library; ~~and~~

~~identifying, for each of the object types, available methods for each of the exposure objects in the remote access object library, the available methods defining a subset of methods in a corresponding base application object in the base application object library; and~~

generating, via an API object generator, the remote access object library from the metadata and the templates.

15. (Previously Presented) The method of claim 14 wherein defining metadata includes, for each exposed object and a corresponding native object in the identified subset in the object model, attributes corresponding to at least one of delayed or immediate translation, object identifier keys, attribute name mapping and attribute type conversion.

16. (Canceled)

17. (Currently Amended) A method for remote invocation of a base object library via a remote access object library corresponding to a base object source model, the remote access object library providing remote exposure of a subset of objects in the base object library comprising:

invoking, via a client application interface, objects in the remote access object library, the remote access object library performing selective exposure of the objects in the base object library;

identifying a corresponding native object in the base object model;

copying, in an object space, the native object, the native object having attributes;

identifying, from metadata defining the remote access object library, for the attributes of the native object, corresponding attributes in the copied object;

instantiating the copied object to a corresponding API object in the client; and

maintaining a link between the API object and the native object, the link providing a dynamic reflection of the native object in the API object such that references to the API object produce corresponding changes in the native object, the reflection operable to allow the client application interface to manipulate the remote object as a realtime, nonduplicative indicator to the native object, maintaining the link further comprising referencing, in a realtime manner, the native object in response to operations to the instantiated API object, such that the operations produce a nonduplicative, atomic result in the native object via the instantiated API object.

18. (Currently Amended) A computer system server for remote invocation of an object in a base object library via a remote access object library comprising:

a processor in a base application server;

a memory in the base application server operable to store the base object library, the memory further including a base application object space operable to store and execute instantiations of the objects in the base object library

an interface including at least one adaptor operable to receive an API object reference to the remote access object library via a client application interface;

-7-

an object mapping table in the memory operable identify a corresponding native object to the referenced API object reference in the base object library, the processor operable to instantiate the identified native object corresponding to the referenced API object in a client object space, and further operable to maintain a link between the instantiated API object and the corresponding native object, the link providing a dynamic reflection of the native object in the API object; and

a build facility, the instantiating by the processor further employing the build facility to:

identify templates corresponding to object types, the object types corresponding to operations for providing the remote API,

define metadata for each of the objects for exposure in the remote API, the metadata identifying, for each of the exposed objects, runtime behavior of the object;

identifying, for each of the object types, available methods for each of the exposed objects in the remote access object library, the available methods defining a subset of methods in the corresponding base application object;

build, via an API object generator in the build facility, the exposed objects for invocation by the client application.

19. (Original) The computer system server of claim 18 wherein the processor is further operable to:

copy, in the base application object space, the identified native object, the native object having attributes;

identify, for the attributes of the native object, corresponding attributes in the copied object; and

populate the corresponding attributes in the instantiated API object.

20. (Original) The computer system server of claim 19 wherein populating further comprises identifying attributes of the native object and determining, from metadata

defining the remote access object library, attribute values of the corresponding instantiated API object.

21. (Original) The computer system server of claim 18 wherein the processor is operable to maintain the link by referencing, in a realtime manner, the native object in response to operations to the instantiated API object, such that the operations produce a nonduplicative, atomic result in the native object via the instantiated API object.

22. (Original) The computer system server of claim 18 wherein the processor is further operable to:

- receive an indication of invoking a traversal method in the API object in the remote access object space;

- identify, via an object identifier in the indication, the corresponding native object in the base application object space;

- determine a related object associated with the native object in the base object library; and

- transmit to a remote access object space, via a traversal service and the object identifier, an instantiation of the related object, the related object corresponding to the association to the native object in the base object library and the instantiation received from a copied object of the related object, the instantiation maintaining a dynamic link to the related object.

23. (Original) The computer system server of claim 22 wherein the object identifier is operable to uniquely identify an instantiation of an object in an object space.

24. (Original) The computer system server of claim 18 wherein the base object library further comprises exposed objects and local objects, each of the exposed objects having a corresponding native object in the object model, the exposed object operable to provide similar operations via the remote application interface as the corresponding

native object, and further wherein invoking includes determining an exposure attribute indicative of whether the invoked object is an exposed object.

25. (Canceled)

26. (Currently Amended) The computer system server of claim ~~18~~²⁵ wherein the base application object library defines a storage area network management application having a database of storage area network management information, the storage area network management application operable to manipulate agents corresponding to manageable entities, and wherein the remote access object library is a toolkit operable to provide API entry points into the storage area network management application in a nonintrusive manner for interrogating the agents and corresponding database.

27. (Original) The computer system of claim 18 wherein the processor is further operable to:

- receive a set of changes modifying the native objects in the base object library;
- and

- remap the links between the API objects and a new corresponding native object resulting from the modifying, wherein the client application interface remains unmodified such that the API object remain executable without rebuilding.

28. (Currently Amended) A computer system application build facility for defining and deploying a remote access object library corresponding to a base application object model comprising:

- a requirements instance manager operable to identify a subset of objects in the base application object model as exposure objects for inclusion the remote access object library, and further operable to define a client application interface accessible to a client application operable to provide accessibility references for the included objects;

- metadata defined by the requirements instance manager, the metadata indicative of attributes for objects in the remote access object library;

an architecture mapper operable to define object templates corresponding to object types in the base application object model, the object templates further comprising:

an identification of object classes for the object types in the base application object model; and

attributes for each of the object types, the attributes operable for definition in the corresponding object in the remote access object library;

metadata identifying, for each of the object types, available methods for each of the exposure objects in the remote access object library, the available methods defining a subset of methods in the corresponding base application object in the base application object library; and

an API object generator operable to generate the remote access object library from the metadata and the templates, such that each of the API objects are reflective of the API object for selectively exposing the corresponding native object.

29. (Previously Presented) The computer system of claim 28 wherein the requirements instance manager is further operable to define metadata including, for each exposed object and a corresponding native object in the identified subset in the base application object library, attributes corresponding to at least one of delayed or immediate translation, object identifier keys, attribute name mapping and attribute type conversion.

30. (Canceled)

31. (Currently Amended) A computer program product having a computer readable storage medium operable to store computer program logic embodied in computer program code encoded thereon as a set of processor based instructions for remote invocation of an object in a base object library via a remote access object library comprising:

computer program code for invoking, via a client application interface, an API object reference in the remote access object library;

computer program code for identifying a corresponding native object to the invoked API object reference in the base object library;

computer program code for instantiating an API object reflective of the identified native object-as-an API object in a client object space, the API object selectively exposing the corresponding native object; further comprising:

computer program code for identifying templates corresponding to object types, the object types corresponding to operations for providing the remote API;

computer program code for defining metadata for each of the objects for exposure in the remote API, the metadata identifying, for each of the exposed objects, runtime behavior of the object, the metadata further identifying, for each of the object types, available methods for each of the exposed objects in the remote access object library, the available methods defining a subset of methods in the corresponding base application object in the base object library; and

computer program code for building, via an API object generator in the build facility, the exposed objects for invocation by the client application; and
computer program code for maintaining a link between the instantiated API object and the corresponding native object, the link providing a dynamic reflection of the native object in the API object, maintaining the link further comprising referencing, in a realtime manner, the native object in response to operations to the instantiated API object, such that the operations produce a nonduplicative, atomic result in the native object via the instantiated API object.

32. (Currently Amended) A computer readable storage medium having a set of processor based instructions encoded as program code on the computer readable storage medium for remote invocation of an object in a base object library via a remote access object library comprising:

program code for invoking, via a client application interface, an API object reference in the remote access object library;

program code for identifying a corresponding native object to the invoked API object reference in the base object library, the base object library defining a storage area network management application having a database of storage area network management information, the storage area network management application operable to manipulate agents corresponding to manageable entities, and wherein the remote access object library is a toolkit operable to provide API entry points into the storage area network management application in a nonintrusive manner for interrogating the agents and corresponding database;

program code for instantiating the identified native object as an API object in a client object space; further comprising:

program code for identifying templates corresponding to object types, the object types corresponding to operations for providing the remote API;

program code for defining metadata for each of the objects for exposure in the remote API, the metadata identifying, for each of the exposed objects, runtime behavior of the object, the metadata identifying, for each of the object types, available methods for each of the exposed objects in the remote access object library, the available methods defining a subset of methods in the corresponding base application object in the base object library; and

program code for building, via an API object generator, the exposed objects for invocation by the client application; and

program code for maintaining a link between the instantiated API object and the corresponding native object, the link providing a dynamic reflection of the native object in the API object, maintaining the link further comprising referencing, in a realtime manner, the native object in response to operations to the instantiated API object, such that the operations produce a nonduplicative, atomic result in the native object via the instantiated API object.

33. (Currently Amended) A computer system server responsive to a set of processor based instructions stored on a computer readable medium for remote invocation of an object in a base object library via a remote access object library comprising:

means for invoking, via a client application interface, an API object reference in the remote access object library;

means for identifying a corresponding native object to the invoked API object reference in the base object library;

means for instantiating the identified native object as an API object in a client object space; the means for instantiating further comprising:

means for identifying templates corresponding to object types, the object types corresponding to operations for providing the remote API;

means for defining metadata for each of the objects for exposure in the remote API, the metadata identifying, for each of the exposed objects, runtime behavior of the object, the metadata identifying, for each of the object types, available methods for each of the exposed objects in the remote access object library, the available methods defining a subset of methods in the corresponding base application object in the base object library; and

means for building, via an API object generator, the exposed objects for invocation by the client application; and

means for maintaining a link between the instantiated API object and the corresponding native object, the link providing a dynamic reflection of the native object in the API object, the API object having a direct reference linkage to a counterpart native object in the base application object space, maintaining the link further comprising referencing, in a realtime manner, the native object in response to operations to the instantiated API object, such that the operations produce a nonduplicative, atomic result in the native object via the instantiated API object.